



# Lakes in Berland

Link submit: <http://codeforces.com/problemset/problem/723/D>

Solution:

C++	<a href="http://ideone.com/lzWoQx">http://ideone.com/lzWoQx</a>
Java	<a href="http://ideone.com/npNMyz">http://ideone.com/npNMyz</a>
Python	<a href="http://ideone.com/Ue9a6Y">http://ideone.com/Ue9a6Y</a>

Tóm tắt đề:

Có một bản đồ kích thước  $n \times m$  với  $n$  là số lượng hàng ngang,  $m$  là số lượng hàng dọc. Mỗi ô  $(i, j)$  nằm trên dòng  $i$  và cột  $j$  sẽ được quy định bằng dấu '\*' hoặc '.'. Trong đó '\*' được quy định là đất, còn '.' là nước. Bao bọc xung quanh bản đồ này là biển.

Một vùng được gọi là hồ nếu như thỏa mãn hai điều kiện sau:

- Vùng này toàn là nước, hai ô nước được tính là kề nhau nếu như chúng chung cạnh.
- Vùng này không giáp biển (lưu ý biển có thể từ biên đi vào trong theo đường nước '.').

Bạn được yêu cầu lấp lại các hồ này sao cho tổng số lượng ô nước cần lấp là ít nhất, và phải còn lại đúng  $k$  hồ.

Input:

Dòng đầu gồm ba số nguyên dương  $n, m, k$  cách nhau bởi dấu cách, đại diện cho số lượng hàng ngang, số lượng hàng dọc và số lượng hồ cần thiết còn lại sau khi lấp ( $1 \leq n, m \leq 50, 0 \leq k \leq 50$ ).

$n$  dòng sau, mỗi dòng gồm  $m$  ký tự. Ký tự thứ  $j$  trên dòng thứ  $i$  thể hiện ô  $(i, j)$  là nước hay đất. Nếu ô  $(i, j)$  là '\*' thì ô  $(i, j)$  là đất, ngược lại là nước nếu như là '.'.

Dữ liệu đảm bảo chứa ít nhất  $k$  hồ.

Output:

Dòng đầu tiên chứa một số nguyên là tổng số lượng ô nước nhỏ nhất cần được lấp đầy.

$n$  dòng sau, mỗi dòng gồm  $m$  ký tự là trạng thái của bản đồ sau khi lấp. Nếu có nhiều đáp án, in ra một đáp án bất kỳ trong số đó.

Dữ liệu đảm bảo luôn tồn tại đáp án.

**Ví dụ :**

5 4 1 ***** * . * ***** ** . * * . **	1 ***** * . * ***** ***** * . **
--	---

**Giải thích ví dụ:**

Ta sẽ lắp một ô nước ở vị trí (4, 3) và bản đồ sẽ ra như hình bên.

**Hướng dẫn giải:**

Ta sẽ có một mảng lưu các hồ nước có thể lắp. Mỗi hồ nước sẽ bao gồm các ô chứa trong hồ nước đó.

Sử dụng thuật toán DFS chạy từ các ô là nước mà ta chưa viếng thăm trước đây. Trong thuật DFS, ta đánh dấu với những hồ nào thuộc biển thì không tiến hành đưa vào mảng lưu hồ nước.

Sắp xếp lại mảng lưu hồ nước theo thứ tự tăng dần kích thước và chọn ra các hồ nước có kích thước nhỏ nhất cần lắp để được k hồ.

**Độ phức tạp:**  $O(n * m)$  với n và m lần lượt là kích thước của bảng.